

VOLUME 32 : ISSUE 6

DEC JAN 2015/2016

QL QUANTA

THE DEFINITIVE QL MAGAZINE



EDITORIAL

NEWS

BACK TO BASICS

SIDE

TSP UPDATE

SPRITE DESIGNER

CHAIRMAN'S NOTES

COPYDATE



QUANTA

INFORMATION ON THE ASSOCIATION

Membership of QUANTA, the independent QL user group, is by annual subscription. The Membership Secretary can supply full details. Copies of the association's constitution & annual accounts are available from the Secretary.

QUANTA Committee - 2015/2016

Chairman: Sarah D. Gilpin 181, Urmston Lane, Stretford, Manchester. M32 9EH

Email: chairman@quanta.org.uk

Tel: 0161 865 2872

Secretary: Alison Southern 40, Distaff Road, Poynton, Cheshire. SK12 1HN

Email: secretary@quanta.org.uk

Tel: 01625 850067

Treasurer & Membership Secretary: John Gilpin 181, Urmston Lane, Stretford, Manchester. M32 9EH

Email: treasurer@quanta.org.uk

Email: membership@quanta.org.uk

Tel: 0161 865 2872

Magazine Editor: Lee Privett. 25 Mucking Hall Road, Barling Magna, Southend on Sea, Essex. SS3 0NH.

Email: editor@quanta.org.uk

Mobile: 07985 513 234

Librarian: David Buckley 136, Denton Lane, Chadderton, Oldham. OL9 8PS

Email: librarian@quanta.org.uk

Tel: 0161 622 1600

Helpline & News Editor: Dilwyn Jones 22 Erw Las, Coetmor New Road, Bethesda, Gwynedd. LL57 3NN

Email: helpline@quanta.org.uk

Email: news@quanta.org.uk

Tel: 01248 601599

Web Administrator & Web Master: Keith Dunbar 44, Dalton Avenue, Stretford, Manchester. M32 9TP

Email: webadmin@quanta.org.uk

Mobile: 07789 483 800

I.C.M.: Dave Park, 1406 Dana Court, Round Rock, TX USA 78664

Email: dave@quanta.org.uk

Tel: (512) 529 7863

Back Issues Requests: Email: backissues@quanta.org.uk

QUANTA reserves the right to publish, reject or edit any material submitted. The opinions expressed are solely those of the contributors. Member's private small ads - free (up to 50 words). Non-members private small ads (and members after 50 words) 50p per 10 words or part thereof. Terms strictly paid up-front with order. Receipt if requested. All copyrights and trademarks are hereby acknowledged.

CONTENTS

EDITORIAL	LEE PRIVETT	4
QUANTA NEWS	DILWYN JONES	7
QUO VADIS DESIGN	ADVERTISEMENT	8
RWAP SOFTWARE	ADVERTISEMENT	16
BACK TO BASICS	JOHN GILPIN	19
JOCHEN MERZ SOFTWARE..	ADVERTISEMENT	22
TSP UPDATE	STEVE POOLE	23
CHAIRMAN'S NOTES.....	SARAH GILPIN.....	24
SIDE PT2	GEORGE GWILT	25
SMALL ADS.....	ADVERTISEMENT	32
SPRITE DESIGNER PT 2	LEE PRIVETT.....	37
RWAP MEMBRANES	ADVERTISEMENT	40
COPY DATE	(FOR NEXT ISSUE)	51
QL FORUM	ADVERTISEMENT.....	52

Visit our website at :

www.QUANTA.org.uk

'Like' us on Facebook at :

www.Facebook.com/QUANTA.org

Wishing all of our QUANTA members and their families a Merry Christmas, a Happy New Year or equivalent for your culture, continent or current location. Well maybe, just maybe that covers all circumstances but you can never hope to please everyone.

I do hope that you all take a rest from QL'ing and have some form of celebration in your household this year. For me it starts slightly earlier with the exciting prospect of the latest Star Wars film, The Force Awakens. Much anticipated by me and after many years of speculation it will have arrived later in December 2015.

This wait has been a long time, ever since the Sinclair QL came on to the scene in fact! The first three films ended with the Return of the Jedi in 1983, just before the Sinclair QL was, ahem! released. It was twenty years later that the next three sequel/prequels were then released but the less said about those three films the better.

Revisiting my younger years should be very exciting in this latest film, as I hope like many others it is a return to the whole of the storytelling and character building the film needs as originally depicted in the first three films (ep. 4, 5 & 6) and with less obvious CGI and more real sets and effects.

—

Another visit to my past occurred when writing the last issue's experiences in the editorial. It was around the time where the ZX81 was currently sold to the public, I had worked with five other colleagues at the Ford Motor Company in Dagenham, in a newly created Electronics Lab covering a three shift pattern, with two people on each shift. We had bought a ZX 81 between us to

experiment on, and it was left in the lab for each shift to use. One particular day one of us had copied from a magazine, a sort of string search program that stored a text file in a string variable that was only limited in size by the available memory (I think). Anyway the following weekend four of us were chatting about this program and its search possibilities when a lightbulb moment happened.

A main part of our role in the Lab was to fault find circuit boards, units, machines etc. from the electronic or programming point of view. PLCs were prevalent then and the Paint, Trim & Assembly plant (PTA) had used three different manufacturers for such purposes as process control (GEC GEM80), conveyor control (Allen Bradley) and overhead systems (Square D).

We logged all faults and the resultant solution in a book, written by hand. The problem when a similar fault occurred we used the log book in trying to find a previous entry.

Well it's obvious today that a database would solve this problem, however databases had yet to come out of the woodwork then. We tested the ZX81 program in storing and searching entries etc. and it worked a treat. However after some use that same day entering faults and their solutions we quickly run out of the 1K of memory.

So we went to see the "Ford Computer Engineer Managers" with a request for a computer and a BASIC programming language (WANG systems as it was then) and put forward our case and justifications (mainly about the speed of troubleshooting existing problems) and after a few questions by them, it appeared we would need a 5^{1/4}" floppy drive as well as a megabyte of memory a year to store the data. This was far too much and expensive, so our request was denied.

So we returned to our lab a little despondent, but we carried on in the mundane ways of repairs, me saving up for a BBC Micro Model B. however, not long after I was permanently distracted by a ZX Spectrum, 1983 ooh look a QL. What if we had have been given that computer?... What if we had used SuperBASIC?....

It was while I was putting the finishing touches to the article about the Sprite Designer that a thought struck me about the purpose of a joystick only controlled program. Yes you can have games and applications (SPACE INVADERS and PAINT as two examples) that work really well with a joystick controlled environment.

Move in four or eight directions, DO or FIRE with the button or combination of the two. However, "I got to thinking", the signals from a joystick are just input signals to the QL that programs can make use of, you just plug in some wires. Now if only there were outputs you could do the same, just plug in some wires!

As of yet, the editor has not received any response regarding the changes to the listings format. This issue we are trying another slightly different style and will await further input. Please have your say.

Old fact. John Gilpin, our treasurer and membership secretary in a recent communique to me enlightened me to the meaning behind the scouts (or cubs) Dib Dib Dib. It's a long story so I wont bore you with all the details but suffice to say, and I assume anyone older than me already knows this, it's actually spelt DYB DYB DYB when shouting in the meetings followed by 'Do Your Best'.

This is followed with the response DOB DOB DOB, we will 'Do Our Best'. I had either forgotten or never known this as a member of

both the Cubs and Scouts. Yet membership of QUANTA and the QL many years later has brought that to my attention. Do old facts eventually happen to all of us I wonder.

—

Finally compliments of the season to one and all and I hope you all have a happy Xmas and a safe and better 2016.

QUANTA NEWS

DILWYN JONES

If you have QL-related news items that you'd like us to include on this page, please get in touch with News Editor - Dilwyn Jones at news@quanta.org.uk

QL-SD Available Again

Paul Veltjens reports that at the time of writing (mid December) a limited quantity of the QL-SD SD card adaptor system for QL is available again through the SellMyRetro.com site:

<http://www.sellmyretro.com/offer/details/QL-SD-internal-SDHC-Card-Interface-for-QL-%28switchable%29-3703>

Two versions of the QL-SD are available, the newer version having a disable switch option to disable the QL-SD interface in order to use the EPROM slot, which is not available while the QL-SD is in use.

QL As USB Keyboard

For a while now, Tynemouth Software have made kits to convert older computers into USB keyboards. We've probably all seen or

QUO VADIS
DESIGN

Independent Information
Technology Services

www.ql-qvd.com

QL/QDOS/SMSQ/E Software

QUO VADIS DESIGN Independent Information Technology Services

QL/QDOS/SMSQ/E Software

Home Products Support Company Contact

Welcome

Quo Vadis Design sells software for the Sinclair Quantum Leap computer (QL) and variants including a new OS called SMSQ/E.

The QL is a computer in its 25th year Anniversary.

News

- QVD QL News Blog - keep up to date
[News Blog](#)
24/04/2009
- Quo Vadis Design Website Launched
21/02/2009

FEATURED PRODUCT



BUY NOW!

Software emulations of the QL now available which run on a PC Mac with Windows or Mac Operating systems.

Copyright ©2009 Quo Vadis Design. All Rights Reserved

Home | Products | Support | Company | Contact | About

Bruce@ql-qvd.com

Quo Vadis Design
38 Derham Gardens
Upminster
RM14 3HA
UK

Tel: +44 (0)20 71930539
Fax: +44 (0)870 0568755

**QPC2 IS NOW FREE AND
AVAILABLE FROM MARCEL
KILGUS WEBSITE**

<http://www.kilgus.net/qpc/index.html>

**Congratulations to Marcel
on becoming a Father**

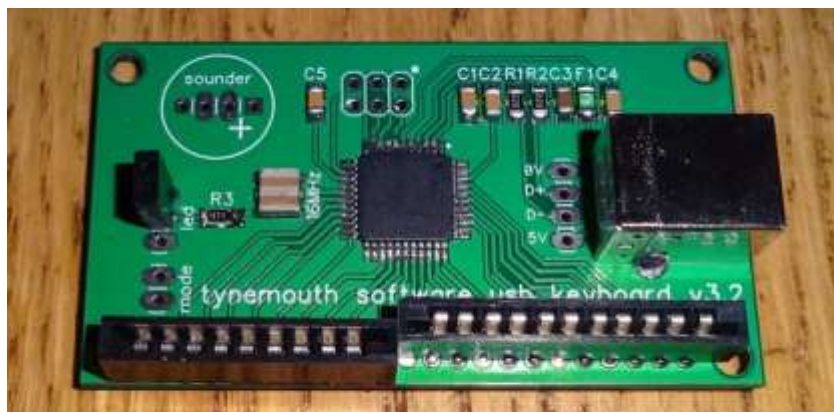
**Check the QL News Blog on
our website for updates.
www.ql-qvd.com/blog**

heard of the Spectrum Recreated, which is basically a Bluetooth keyboard driving a Spectrum emulator app on the host system. This Tynemouth Software kit takes an original computer and converts its keyboard into an USB keyboard.

In the case of the Spectrum kit, this comprises of a replacement keyboard membrane and USB controller board – the original computer board and keyboard membrane are removed and replaced with the parts from this kit. A good use for a computer with a failed motherboard, for example.

Following enquiries by people like QL Forum’s Peter Scott, Tynemouth have made available a QL version of the kit. They have posted an article about the QL kit here:

<http://blog.tynemouthsoftware.co.uk/2015/12/day-3-sinclair-ql-usb-keyboard.html>



The USB Controller Board

Peter Scott reports that as he ordered two kits, the price was £60 each plus £5 postage. The cost reduces the more kits are ordered,

making group buys attractive. If you have queries, Dave Curran of Tynemouth Software is on QL Forum under the moniker “tynemouthsw”.



The keyboard membrane plugged into the USB controller board mounted on the QL keyboard back=plate

Tynemouth Software have a shop on the Etsy shopping site:

<https://www.etsy.com/uk/shop/TynemouthSoftware>

SMSQ/E v3.23

SMSQ/E Registrar Wolfgang Lenerz has announced the release of version 3.23 of the operating system SMSQ/E.

Wolfgang says that “This is just a minor bug fix in some of the DMedium_XXX keywords.” The binaries and sources may be downloaded from:

<http://www.wlenerz.com/smsqe/>

SMSQmuLator

SMSQmulator version 2.13 is out.

It comes with the usual train of bug fixes (you should upgrade), but also with an Italian version thanks to Giorgio Garabello.

If you have never come across SMSQmuLator before, it is a QL emulation for computer platforms which use the Java system. See www.java.com for information about Java.

As with previous versions, SMSQmuLator comes in versions for Java 6, 7, or 8, depending on what is installed on your system. If in doubt, download SMSQmuLator for the higher version number first and try that.

A list of changes in each version may be seen at :

<http://www.wlenerz.com/SMSQmulator/version.html>

Download SMSQmuLator free of charge from :

<http://www.wlenerz.com/SMSQmulator/>

Q68 News

Peter Graf wrote on the QL-users list back in November about the latest news on his proposed Q68 system. I quote from his email here.

“After demonstrating the Q68 at the Edinburgh meeting, I made a hardware redesign, so the Q68 board is now easier and cheaper to build. Also I did some work toward implementing a data/instruction cache. As for Q40/Q60, the Q68 can not reach them in terms of speed. At the moment, the Q68 is only slightly faster than

the QXL. However, since it is FPGA based, a later generation could benefit from faster chips without the need for a completely new design. If I remember correctly, the FPGA generation I picked for the Q68 is already 8 years old, so there is room for more speed, should there be new generation Q68 someday.

The Q68 has 32 MB SDRAM @ 100 MHz, high-colour graphics, Ethernet, two SDHC slots, PS2 keyboard and mouse, stereo sound, SER, I2C, battery buffered real-time clock. I didn't find time for QL work lately, but I still hope to bring the Q68 into a stage where it is useful for others.”

Black Phoenix News

Giorgio Garabello writes:

Black Phoenix works on SMSQmulator now (not only QPC2)

www.quantum.altervista.org

News Editor's note: Black Phoenix is a distribution in QXL.WIN format of a large collection of QL freeware software.

Trump Card Clones from Russia

Rich Mellor writes:

“I see that someone from Russia has now cloned the Trump Card and is offering them for sale (they have a few in stock at the moment):

<http://www.sellmyretro.com/offer/details/Sinclair-QL-RAM-expansion-and-floppy-disk-interface-10861>

At last someone has managed to make a replica J

"tetroid" in Novosibirsk, Russia, Russian Federation , selling for £69.90+delivery from £10. Payment by Paypal”

Forth Email List

An email list has just been created to discuss the Forth programming language on Sinclair computers: Forth on Sinclair (and Z80) computers.

<http://programandala.net/en.forth-sinclair.html>

Any QL user with an interest in Forth is welcome. In the Spanish QL users group there were three of us, including me (Marcos Cruz), that programmed the QL in Forth. By the way, there's also a new email list to discuss Forth in Spanish:

Forth ES: <http://programandala.net/es.forth-es.html>

The Distribution v3.03

Urs König writes:

The current “completed download” count for THE DISTRIBUTION since release day 2015-04-11 is 240! A slightly improved version (3.03) has been released during November. Major changes are related to new versions SMSQmulator and SMSQ/E. For further information about The Distribution, have a look at:

<http://www.sinclairql.net/repository.html>

Note that the download file size for the full version is a massive 3.35GB (yes, gigabytes – that’s big by QL standards), although if you have already downloaded v3.01 or later, there is a cumulative download updater file which is “only” 81MB in size!

QL/E v3.11

A companion distribution from the same stable, QL/E is described as “a modern and compact ready to run QL environment”. It is supplied as a bootable .WIN format container for use with SMSQ/E systems. Version 3.11 was released in November 2015. Note that it is included in The Distribution (see above) so if you already have that, you don't need to also download QL/E but QL/E is available to download as a separate stand-alone system if you wish, also from <http://www.sinclairql.net/repository.html>

Software Updates From George Gwilt

George Gwilt's website www.gwiltprogs.info contains two new items.

1. Turbo Toolkit TURBO_TK v 3.44

This allows proper configuring, by CONFIG or MENU_CONFIG, of the default maximum length of string for the function EDIT\$.

2. In GDLIB lex5 is a slightly improved version of the subroutine which creates and activates an executable program. This version allows the setting of registers D0 - A3 in the activated program. It also returns the ID of the program.

Qubide News

Jose Leandro has announced that the newer versions of the Qubide can have a through connector (see picture). One downside to this is that the IDE connector faces upward from the board, meaning that if you wish to plug in a Compact Flash or SD card converter board, it will end up standing up at right angles to the board.

This is clearly impractical where the Qubide is plugged into an original QL case, so Jose has come up with a compact right angle connector board which turns the flash memory card horizontal, as you can see in the picture.

Those wishing to buy a Qubide can contact Jose Leandro via QL Forum (use the moniker jose_leandro) – details on page 33 of the August/September issue.



Picture of the new Qubide with through connector (right) and right angle adaptor board (centre) for the Compact Flash adaptor card (left)

Dr Carlo Delhez

Bob Spelten jr. informed us of the death of former QL programmer Dr. Carlo Delhez in mid October:

“This weekend I came across the sad announcement that Dutch programmer Dr. Ir. Carlo Delhez had died a week before. He was only 48. He was a familiar name among Sinclair users. He wrote

RWAP Software

We have been working hard to safeguard the future of the Sinclair QL through creating an online QL Wiki backed by a repository of QL titles.

The QL Wiki enables people to see what software and hardware was produced for the Sinclair QL. The repository enables people to purchase working copies when their microdrive cartridges fail, or they move onto disks or emulators. Where we can, we obtain permission from the copyright holders to re-release items on the market.

This has enabled us to bring some of the titles back to the market as well as re-kindle interest in the Sinclair QL and its emulators.

Old favourites brought back to life include:

E.V.A., Vroom
Black Knight, Double Block
Jungle Eddi, The Lost Pharaoh
Pangl, the Lonely Joker
Microdeal's Flight Simulator, QL Hopper, The King,
Night Nurse and Cuthbert in Space
Mortville Manor, Darkside of the Moon [£10]

We still retain our existing catalogue, including:

Open Golf, Return to Eden,
Stone Raider II, The Prawns,
Hoverzone, Lost Kingdom of Zkul,
Deathstriks, Nornaday,
Flightdeck and QWord

All games are available on disk or for use with Q-emulator on the PC **from ONLY £5 each**

Microdrive versions also available - from £10 each.
Memory / system limits may apply - please check before ordering.

Visit the QL Wiki for more details on software, books and hardware for the Sinclair QL:

www.rwapadventures.com/ql_wiki

RWAP Software
3 Dale View Court, Fulford, Stoke-on-Trent
Staffordshire ST11 9BA
Tel: 01782 398143
Email: rich@rwapservices.co.uk

Website:
www.rwapsoftware.co.uk or www.sellmyretro.com



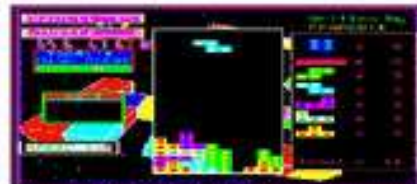
Vroom by Pyramide



E.V.A. by Westway



Mortville Manor by Pyramide



Double Block by Francois Lanreut



Night Nurse by Microdeal

programs for ZX & QL computers, both commercial and freeware. Among them Coral Basic - an extension bringing SuperBASIC to the ZX81 - and ZX emulators for the QL and PC. His Ungif tool has seen regular use on my systems. He was not a club member nor a regular visitor at meetings but I remember talking to him in Eindhoven briefly, about 20 years ago, a very nice person and full of new ideas.”

Several prominent QLers were quick to add their memories. Geoff Wicks wrote:

“This is indeed sad. I never met Carlo, but I did have contact with him sending him Dutch and English word lists for his QLingo program, way back in 1995. He was probably not so well known outside the Netherlands, but he did produce some interesting software.”

Darren Branagh wrote:

“Very sad to hear this...a young man. I used Carlo's emulators a lot when I first got a QL, as I moved from a Spectrum. He was always very helpful, although like Geoff I never actually met him.”

Our thoughts are with Carlo's family.

Wolfgang Lenerz Backup

Wolfgang Lenerz has released an update to his Backup program. V1.07 corrects an issue where it didn't handle Prowess types and Proforma drivers correctly - <http://www.wlenerz.com/qlstuff/>

Prospero Pro Fortran Manual

The Pro Fortran package for the QL has been available as

freeware for a while now, but we were missing a manual. Now, thanks to Julian Skidmore, we have been able to make the manual available as a PDF file for free download. Note that the PDF is 67MB in size. Download the manual on the following page – scroll down to “profortran.zip”, the link to the manual is just under that.

<http://www.dilwyn.me.uk/language/index.html>

JS ROM Disassembly

I have been able to add another version of the JS ROM disassembly to my website, thanks to David Westbury. This adds to the disassemblies and sources for other versions of the QL operating systems. Download it from:

<http://www.dilwyn.me.uk/language/index.html>

GST QC

John Powell has kindly sent me a disk image version of the GST QC compiler, together with a zip file of screen dumps of the compiler in action. Download both from:

<http://www.dilwyn.me.uk/language/index.html>

Vision Mixer 2 Update

Following a suggestion from a user in Italy, Vision Mixer 2 presentation software has been updated to allow the filename of a VMC (Vision Mixer Control) file to be passed in an EX command, e.g. direct via File Info II if you use that software.

Download version 1.01 from :

<http://www.dilwyn.me.uk/graphics/index.html>

Think back to 2014. While QUANTA was celebrating it's Thirtieth birthday in Edinburgh, NEMQLUG (North East Manchester QL Users Group) were celebrating it's coming of age. It's hard to realise that it is Twenty-one years since this QUANTA Sub-group first got together in the New Moston Library Meeting Room. Those were the days when nearly all programmes for the QL were written in SuperBASIC and our sub-group was no different from many others.

While we (the members of NEMQLUG) were wondering what we could take to Edinburgh with us that was within our capabilities, we decided that as our Birthday celebration we would return to the beginning and see if we could write something in SuperBASIC that everyone could join in with – anyone with a Black-Box QL that is. – and others who have advanced, but feel like a BASIC challenge.

We had talked for a long time about putting together a Games CD just for the QL but came to the conclusion that just mere talking about it wasn't getting us any nearer to our aim. It was time to act!!

Our first game, a simplified version of the game "2048", was suggested and after further discussion, it was agreed that John Southern would master mind the project by pulling to bits an existing game off the Internet and write a basic plan as to how to proceed. Within the first week or so, we had a very basic version written in SuperBASIC and reduced to the minimum number of lines of code just as an exercise to see how condensed it could be, before we were satisfied with our work,

John decided that if we were to pass it onto those QUANTA

Members who still only had a Black Box QL, it would need quite a bit of expanding – not only to alter any of the code but as an explanation of the thought processes that went with it and a number of explanatory REMark statements were added.

Fast forward to the present day. I, as Membership Secretary & Treasurer of QUANTA am responsible for notifying our members when their subscriptions fall due, and to reduce my work load around the Christmas/New Year period, I added to my “Subs Due” letter a request that as many people as possible might consider sending their subs as early as possible so that I could deal with them and spend more time with my family over the festive season.

As usual, those members who have no Internet etc. got out their cheque books and together with their subscription remittance sent me a note often to thank the QUANTA Committee for their work on behalf of all the QUANTA members and others who added a few comments about the magazine.

I realised that many of our more elderly members still only have a BB QL, and comments like “ I find a lot of the articles in the magazine are way over my head” OR “ I find little of interest in the magazine any more as the articles have followed (very reasonably) the advancement of the QL system but occasionally I can find something of interest to read”.

I reported back to QUANTA and the sub-group on this situation and it was almost immediately agreed that during 2016, we would send Lee (The magazine editor – who is always crying out for further copy) three of our games that we have been working on.

The idea is to start off in the February/March issue with the

SuperBASIC version and lots of John Southern's thought processes that went into this game. As an extra, having got John (S) 2048, the display was extremely BASIC and I offered to try and add some Easy Pointer 4 code to bring the game into the pointer environment and to add a bit of polish to the resulting output.

This additional code will be dealt with in a second article in the April/May issue of the magazine. This will be repeated in the following 2 issues with details of our second game and finally to end off the magazine year, our third game – BASIC in the Oct/Nov issue and any advancement to this in the Dec/Jan magazine.

All this will prove that the SuperBASIC language is nowhere near dead, and anyone who still has a BB QL as their main machine is invited to offer improvements to our code or even have a try at writing a new game of their own.

I'll bet that there are still quite a number of QLers who still have skills in using their QLs and can easily knock a bit of code together. Backtracking to the GREAT Edinburgh Workshop, we presented our version of 2048 and one European member became quite addicted to it.

I hope that those of you who feel left out because you still only have your BB QL will rise to the challenge and any problems that you have in trying to get started or maybe later on, please pass your difficulties on to Lee (contact details inside the front cover of your magazine) and NEMQLUG Members will put our heads together to come up with a solution for you.

You now have a couple of months notice for you to not only enjoy your festive season but also to polish your keyboards and to find

JOCHEN MERZ SOFTWARE

Kaiser-Wilhelm-Str. 302
47169 Duisburg, Germany

Fax +49 203 502012
EMail: SMSQ@J-M-S.com

SMSQ.J-M-S.COM

Thanks to Marcel, QPC2 is now freely available.
You can download it from Marcells homepage

www.Kilgus.net

If you wish to print from QPC2, then you need

QPCPrint

... which is available from J-M-S.

Only 39.90 if you choose EMail delivery.

For additional 4 EUR, delivery will be on CD.

QPCPrint will allow you to print to (more or less) every printer which is installed under Windows (dot matrix, ink, laser, PDF "printer", FAX "printer" etc.)

You can place your order via letter or use the SSL order form on SMSQ.J-M-S.COM - click on "Online orders".

If you order by Mail or through the SSL contact form: We now accept VISA, MasterCard, Diners Club, JCB, Discover, UnionPay, BCard, DinaCard and American Express!

The order form has not been updated for the new card types at the time I write this ad, but it will be updated soon.

I am still working on changing the forms and implementing SEPA on all the order forms.

your QL SuperBASIC operations manual and prepare for some work. I look forward to hearing from you all. Have a GREAT Christmas and let's hope that the New Year 2016 will hold all that you expect from it.

TSP UPDATE

STEVE POOLE

The Travelling Salesman Project team are happy to announce that they have been joined by John Powell, a new Quanta member, who proposed his services to transpose SuperBASIC programs into C, and eventually C++.

Our first objective was to transcode a simple program, in this case a tally_sort, and to time it in relation to its SuperBASIC original. John first rewrote the SuperBASIC program to make it 'C like', and enclosed the program in a for-loop to run 60 times, to make timings sensible.

Then he rewrote the C-like basic in real C, compiled it and executed it. The results were as follows :

C-like SuperBASIC	on QPC2	: 40 seconds	: *1
C-like Turbo compiled SB	on QPC2	: 9 seconds	: *4.5
Compiled C program	on PC	: 0.157 seconds:	*250
Original SB Assembler tally	:	n.a.	: (*240 : on a bare QL)

John gave up on C68, and used a native PC 'C' compiler. These results indicate that PC 'C' programs are somewhat faster than programs using QL-assembled ones, if the Basic is written C-like and is fully optimised to be compiler-compatible.

This confirms that C++ is a fast but verbose language, and that now transposition of our 'Travelling Salesman Program' will be a worthwhile venture.

We have an expert C++ programmer to help us, but the task could take some months to complete. When once our 'shrink' TSP will be in C++, we will be able to compare it to existing TSP programs, (which are all written in C or C++). Also we will be able to use the existing public domain C++ libraries to improve our shrink program, for example by converting the 'Convex Hull' routines into SuperBASIC for backwards compatibility.

As you see work is progressing steadily.

CHAIRMAN'S NOTES

SARAH GILPIN

To all our members; the Committee and I wish you all a very Happy Christmas and all the best for the New Year 2016. We intend during 2016 to try and add some magazine content for those with BB QL's. We are also adding early Newsletters onto the website, the main difficulty being getting them scanned and checked as the old type fonts used are not always easy to read.

Once again we are looking towards the Annual General Meeting 2016, which will be an electronic meeting 181, Urmston Lane, Stretford, Manchester, M32 9EH. A confirmed date has not yet been agreed. Anyone who wishes to attend in person is most welcome. If they confirm that they will be here and can arrive by 12.30pm I will lay on sandwiches.

Full instructions on how to join us will be in the AGM booklet sent out with the Feb/March 2016 magazine. We hope that this will enable some of our overseas members to join us, as well as some members in the UK who do not like to travel.

The Committee would like to thank NEMQLUG for putting on a workshop at the end of September, which although not overly attended was very busy. An idea was put forward by NEMQLUG and during 2016 they are going to hold a number of meetings where ideas and projects can be discussed, or to use the modern terminology “networking sessions”. Further details are in the Sub-Group information pages. We hope that this will be a success and continue into 2017.

SIDE PT.2

GEORGE GWILT

In the previous article I described in outline, how one program, Side1, could set up or remove a second program, Side2, depending on the position of the pointer. Here I will describe Side1 in more detail.

The first step in any pointer program is to set up and display its window. In the case of Side1 its window is a very small black window with no shadow, no border and no contents. It has no sub-windows and no loose items. However, this window has to be set up and the code to do this is shown now.

```
*** Setting up the tiny window ***  
; sidel_asm  
  
in          win1_lib_hed1
```

```

bra.s      start
dc.l       0
dc.w       $4afb
hed1       <"side1_1.01">,fname
ds.w       0

```

```

in         win1_ass_pe_keys_pe
in         win1_ass_pe_qdos_pt
in         win1_ass_pe_keys_wwork
in         win1_ass_pe_keys_wstatus
in         win1_ass_pe_keys_wman
in         win1_ass_pe_keys_wdef
in         win1_lib_hed1

```

```

                Rset      0
id             rs.l      1
wmvec         rs.l      1
slimit        rs.l      1
sysvar        rs.l      1
jobid         rs.l      1    address of lex'd job header

```

```

*
start         lea        (a6,a4.l),a6            dataspace
              moveq     #mt_inf,d0
              trap      #1
              move.l    a0,sysvar(a6)
              bsr       ope                    open a con
channel . . .
              move.l    a0,id(a6)              .. keep the ID
              moveq     #iop_pinf,d0

```

```

        moveq    #-1,d3
        bsr     tp3
        bne     sui      ---->      ptr_gen not
present
        move.l   a1,wmvec(a6)      keep WM vector . .
        beq     sui      ---->      .. wasn't
there!
        movea.l  a1,a2              set WM vector in A2
        lea     slimit(a6),a1
        Moveq   #0,d2              this must be
zero
        Moveq   #iop_flim,d0       maximum size of
window..
        trap    #3
        subi.l  #$C0008,(a1)      .. less 12, 8
        lea     wd0,a3             address of
window definition
        move.l  #ww0_0,d1         size of working
definition..
        Bsr    getsp              sets ALCHP'd
address to A0..
        movea.l a0,a4             .. and to A4

```

```

; We need to set the status area to zeros
; and the loose items to "available" (zero)

```

```

        Lea     wst0,a1           Status ..
        movea.l a1,a0            .. area ..
        Moveq   #wst0_e-wst0-1,d1 bytes to clear - 1
st1 clr.b    (a0)+
        Dbf     d1,st1

```

```

    movea.l  id(a6),a0          Replace the channel ID
    move.l   wd_xmin+wd_rbase(a3),d1
                                minimum size
    andi.l   #$FFF0FFF,d1      Lop off the scaling
factors
    jsr      wm_setup(a2)      Set up the working
definition
    move.l   #$400040,d1      Set the window . .
    jsr      wm_prpos(a2)     . . away from the edge
    jsr      wm_wdraw(a2)     Draw the contents

```

The above code is quite standard apart from one instruction. Since the program is designed to react when the pointer sprite hits the left hand edge of the screen, I arranged that its small window would not be set up near enough that edge to provoke an immediate response. Thus the third last instruction

```

    move.l   #$400040,d1

```

is included to place the window away from the edge.

The next set of instructions is designed to monitor the position of the pointer and to allow the program to continue only when the pointer has hit the left hand edge and has rested there without moving for at least two seconds.

When D2 contains 64 the routine reading the pointer returns when the pointer sprite reaches an edge. To find out which edge we have to examine the position of the pointer, which is in D1.L. Two facts are relevant here. Firstly the position of the pointer will not in

general be at the place where the sprite meets the screen's edge.

Secondly we have no idea what the current pointer sprite is: it may be any of several depending on what programs are loaded at the time. The best we can do is to test whether the pointer is reasonably close to the left hand edge, which I have taken to be within 28 pixels of the edge.

Someone following closely may object to this on the grounds that it could be that the sprite was on the top edge and only quite near the left one. He, or she, would be quite correct in that objection.

However, I decided that a more accurate determination of the actual edge would be pointless. If the pointer is held for over two seconds as close as 28 pixels from the left edge it matters not whether the reading routine returns because the sprite hit an edge other than the left. The result - the appearance of Side2 - would be as expected.

On return from reading the pointer, D1.L contains the horizontal pixel position in the top word and the vertical position in the low word.

```
***** Determine whether to load Side2 *****
```

```
addq.w #8,a1      ---> pointer record
wrpt moveq #64,d2  pointer sprite at edge?
Moveq #-1,d3      timeout
Bsr rdpt          read pointer until edge
Bne sui          ----> terrible mistake!!
move.l d1,d7      is pointer near . .
Swap d7           . . enough . .
```

```

    cmpi.w    #28,d7        . . left edge? . .
    Ble      wrpt1         . . yes

    Moveq    #-1,d3        wait till pointer. .
    Bsr      rmove         . . moves then . .
    Bra      wrpt          . . wait again for edge

```

```

; left edge found

```

```

wrpt1    moveq    #100,d3    wait for 2 seconds
         Bsr      rmove
         beq      wrpt      not long enough - try
again

```

```

*****

```

Now that the pointer has been detected remaining for long enough at the left edge we have to set Side2 going. This is done in the following code by calling lex5 with the appropriate parameters set.

The directory holding Side2 can be set by a config block. The name of the directory is found in stra0 and this is appended to "side2" placed in "totname".

This occurs in the first eight instructions. These include the routines qstrcpy, which copies a string, and qstrcat, which appends a string, these addresses being supplied on the stack. The routines are part of lex5 as stated in the previous article.

Load Side2

; EX side2

```
lxreg      reg      d1-3/a0-1
movem.l   lxreg,-(sp)
lea       fnme,a2      "side2_bin"
and . . .
pea       stra0      . . its
directory
pea       totname
bsr      qstrcpy      copy directory
to totname
move.l    a2,4(sp)
bsr      qstrcat      add prog name
lea      8(sp),sp     clear stack
lea      totname,a0   prog name for
LEX5
suba.l    a2,a2      No register
string
suba.l    a1,a1      No parameter
string
moveq     #0,d1      owned by BASIC
moveq     #32,d2     priority
moveq     #0,d3      EX
bsr      lex5        D1 is the
called . .
move.l    d1,jobid(a6) . . job's ID
movem.l   (sp)+,lxreg restore
registers
```

SMALL ADS

TIRED?

Tired of not seeing your name in print? Then spend an hour (MAX) in doing something to change that. We will put glitter on it! We will publish it. Comments, Programming, Reviews. Offers in excess of 50 words to the Editor

!!!Worried???

Don't worry about your spellin or grandma not being up to scratch, that is somfink we can correct or leaf as necessary, the Editor's decision is vinyl. Why not right a review, A review of a peace of hardware u have recently scene, obtained or ordered that is related or used wiv the Sinclair QL. A review of a peas of software you ave bought or downloaded legally from a website that doesn't stick a virus on your system. Reviews of any uvvar media such as books, magazines or even leaflets that other people may appreciate. Contact Mr Privett for details on how you can overcome ur worry without counselling.

WANTED

With FTC (Fleet Tactical Command) 1 having being found, I'm now trying to track down FTC II by Diren which is currently missing in action. I'm also after Top Team by Arunsoft which was a football management game. If you can help please contact Peter at:

peetvanpeebles@yahoo.co.uk

LOST

The ability to do something positive, then submit something to the editor, you know it makes sense.

= URGENT =

For the magazine are seeking a volunteer for the role of Treasurer. We are also seeking a volunteer for the role of Membership Secretary. Or we are seeking a volunteer for the role of both, please contact the Chairman of QUANTA via the contact page inside the front cover. We will be glad to hear from you

SHEDS SHEDS SHEDS

Shed loads of space for you to write or submit that SuperBASIC listing for the magazine. **Short** Listings - any number of lines really, from just a few to a couple of pages, single or multiple procedures and functions. As long as you have written to either solve or explore a programming problem. Or even just for a bit of experimental fun. **Long** listings? No program is too big, no language too strange, whether its just concepts, SuperBASIC, block diagrams, Boolean logic, assembler, mnemonics (that's one for you George), C# C++ C- - or just plain C, Pascal, Fortran or even S*BASIC, anything about everything would be appreciated. What if you don't know if your program listing is too long to be short or too short to be long? **NO WORRIES**, we accept medium listings too throughout the year so get writing **NOW!**

Contact us in the usual way, so contact us **NOW!**

If you would like to place a small ad then go to page 2 for full details of how to go about it, options are for QUANTA members as well as non-QUANTA members.

We now wait until the pointer is far enough to the right of Side2's window before deleting Side2.

***** Delete Side2 *****

```
Lp1      bsr          susjb      Wait for 3 seconds
         moveq       #-1,d3
         moveq       #48,d2      Find the current . .
         bsr         rdpt       . . pointer position
         move.l      d1,d7
         swap        d7
         cmpi.w     #42,d7      42 is width of Side2's
window
         ble         lp1       still inside
         bsr         rjb       delete side2
         bra         wrpt      start all over again
```

Now follow some subroutines:

"rmove" waits until the pointer moves from the position in D1.

"rdpt" reads the pointer and returns when the condition set in D2 is met.

"tp3" calls Trap #3 and tests D0.L

"susjb" suspends Side1 for three seconds.

"rjb" removes Side2.

"ope" opens a con window.

```
rmove      moveq      #8,d2          move
rdpt       moveq      #iop_rptr,d0
tp3        trap       #3
           tst.l      d0
           rts

susreg     reg        d1/a0-1
susjb     movem.l    susreg,-(sp)
           moveq     #-1,d1          this job
           move.l    #3*50,d3       3 seconds
           suba.l    a1,a1
           moveq     #mt_susjb,d0
           trap      #1
           movem.l   (sp)+,susreg
           rts

rjb_reg   reg        d1-3/a0-3
rjb       movem.l    rjb_reg,-(sp)
           moveq     #0,d3          no error

code      move.l     jobid(a6),d1    job ID
```

```

        moveq    #mt_frjob,d0
        trap     #1                remove Job
        movem.l  (sp)+,rjb_reg
        rts

*

con      dc.w    3
        dc.b    'con'

ope      lea     con,a0            To open "con" .
        .
        moveq   #-1,d1            . . for this
job

        moveq   #0,d3
        moveq   #io_open,d0
        trap    #2
        rts

totname  ds.w    13                space for LEX
name

        hedl    <"side2_bin">,fnme

```

```

/* Configured Items - Config Level 1

```

```

*
* Name      Description
* STRA0     program directory
*/

```

```
DC.L    '<<QC','FX>>'
DC.B    '01'                ; Level
DC.W    0
DC.W    6
DC.B    '1.01  '           ; Version Number
DS.W    0
```

; Strings

```
DC.B    0,0
DC.W    I_STRA0-*          ; -> Item
DC.W    0                  ; -> pre-processing
DC.W    0                  ; -> post-processing
DC.W    D_STRA0-*         ; -> Description
DC.W    A_STRA0-*         ; -> Attribute
DC.W    -1                 ; end of item list
```

; Descriptions

D_STRA0

```
DC.W    17
DC.B    'program directory'
DS.W    0
```

; Attributes

A_STRA0

```
DC.W    $0200
```

; Items

I_STRA0

```
DC.W    12
```

STRA0

DC.W 5
DC.B 'win1_'
DS.W 0
DS.B 7
DS.W 0

```
lex5      in      win1_lib_lex5_sym_lst  
          lib      win1_lib_lex5_bin  
  
          in      win1_ass_pe_sidewl_asm  
  
          in      win1_ass_pe_peas_sym_lst  
          lib      win1_ass_pe_peas_bin  
          in      win1_ass_pe_csprc_sym_lst  
          lib      win1_ass_pe_csprc_bin
```

SPRITE DESIGNER PT. 2

LEE PRIVETT

Welcome to part two of the Sprite Designer Program. In the last issue we started describing the differences between the visual look of the modified program and the original.

As a refresher, we have included a screen shot of the original program followed by the latest modified version.



Original

I should also point out that for the benefit of the members differing systems, this program was written on QemuLator with 128K memory, the Back ROM was Toolkit 2 but it is not needed for the program to run, and the Turbo Toolkit was also loaded in to the same 128K memory but again is not needed for the program to run, however some keywords are used for the editing facilities at the end of the program listing.

This program therefore, should run on a BBQL, it would be very helpful if the QUANTA members with the black box original QLs could test this program on their systems and report back to the QUANTA editor, their findings and any issues for the benefit for all. Additionally, in a future issue there will be a program that will test all the features of the sprites created by this designer, watch this space.



Modified

We left the description with the additional four key presses now available B C F R when in the Editing Sprite window. In the centre window, the Sprite Selection, you can see all 20 sprites available and what they look like actual size with their associated number. 00 is highlighted in yellow and this indicates that this sprite is currently selected.

In order to get the cursor across from the Editor Window to the Sprite Selection window you can do one of two things. Pressing the TAB key or moving across with the cursor key, once the flashing * gets to the rightmost side of the Editor Window, a further 'cursor right' press will move the cursor across.



Cursor in the Sprite Selection Window

RWAP Software

Replacement QL Keyboard Membranes

Special Offer for Quanta Members
£12 each plus post and packing

Quote "Quanta 2012 Offer"

Post and packing:
UK £2.50
Europe £2.80
Rest of World £3

Visit the QL Wiki for more details on software, books and hardware for the Sinclair QL:
www.rwapadventures.com/ql_wiki

RWAP Software
3 Dale View Court, Fulford, Stoke-on-Trent
Staffordshire ST11 0BA
Tel: 01782 398143
Email: rich@rwapservices.co.uk

UK Cheques payable to 'R. Mellor' or use PayPal.

Website:
www.rwapsoftware.co.uk or www.sellmyetro.com

Why a bad keyboard membrane could change your whole career strategy!

Working to safeguard the QL for the future. www.rwapsoftware.co.uk

You can now see that the asterisk disappears from the editing window and the 00 is highlighted in blue, this represents that the cursor is present at 00 and that 00 was also previously selected. In the Commands window, red text on a white background has appeared confirming that that is where the cursor is and the saved sprite 00 is selected.

If cursor is moved using the arrow keys, this text then changes accordingly using the following 'Message' routine.

```
DEFine PROCedure Message
  OVER#5,0
  AT#5,10,0 : PAPER#5,7 : INK #5,2
  Spt=(sprx*10)+spry
  PRINT#5,"Sprite      Number - ";Spt;" "
  AT#5,12,0
  IF t=1 OR Spt=sprn THEN
    PRINT#5," Selected  "
  ELSE
    PRINT#5,"Highlighted"
  END IF
  OVER#5,-1
END DEFine
```

'Spt' is found by the x position of the cursor * 10, this is added to the y position, printed then checks to see if selected already and prints the condition accordingly. Thus x is either 0 or 10 and y is 0 to 9 covering the numbered sprites from 0 to 19 (20 Sprites). 00 to 09 is used on screen to keep formatting aesthetically constant. I did debate whether to add 'Sprite No' or 'Sprite Number', the latter won, however the layout might be better with the former.



Cursor at 01 in the Sprite Selection Window

The previous screen shot now shows the cursor away from 00 and highlighting 01 in cyan and 00 returns to its previous yellow highlight indicating this sprite is selected. The text in the commands window now indicates sprite 01 has been highlighted.

The options displayed at the bottom of the screen are cursor keys, Enter Tab and Shift Tab. These allow the movements of the cursor in the sprite window, cursor to the commands window, cursor back to editor window.

By pressing the Enter key, it selects whatever sprite number is highlighted by the cursor at that point. Text and highlighting change again as appropriate. By pressing the TAB key, the cursor is then moved to the commands window, this can also be achieved by the cursor keys. Pressing SHIFT & TAB in the sprite selection window will return the cursor to the editing window. This again can also be achieved with the cursor keys.



Cursor Tabbed to Commands Window

The screen shot above shows the cursor in the commands window LOAD is highlighted and text appears below it as a prompt for the action. The difference between Load & Save and Store is that the former refers to the sprites and memory and the latter is about saving the complete file to the default drive, which is the one used by the program when it is RUN. To make further file sets of sprites would require further changes to the program or copying the actual file to another folder/drive or renaming.



Cursor Highlighting Store Command and Text Prompt

Part 1 of the full listing is shown as follows:

```
100 REMark -----
110 REMark SPRITE EDITOR
120 REMark Originally by Herbert Zenz
130 REMark Modified slightly by
140 REMark Lee Privett 2015 to use the
150 REMark following keys:
160 REMark CURSOR ESC SPACE ENTER TAB F B
170 REMark -----
180 :
190 REMark -----
200 REMark Default Drive String DD$
210 REMark -----
220 :
230 DD$="FLP5_"
240 :
250 anf=RESPR(7200)
260 LBYTES DD$&"sprsys_bin",anf : CALL anf : anf=anf+4
270 :
280 SpriteEdit
290 STOP
300 :
310 REMark #####
320 REMark Set up for TV screen values
330 REMark #####
340 :
350 DEFine PROCedure INITtv
360 mo=8 : st=1 : xm=7 : po=3 : Ste = 2
370 END DEFine
380 :
390 REMark #####
400 REMark Set up for Monitor screen values
410 REMark #####
```

```

420 :
430 DEFine PROCedure INITmon
440   mo=4 : st=2 : xm=15 : po=7 : Ste = 1
450 END DEFine
460 :
470 REMark #####
480 REMark Main Program Calls
490 REMark #####
500 :
510 DEFine PROCedure SpriteEdit
520   INITing
530   REPEAT Main
540     EDITing
550     SELECTing
560     IF sprx=-1 : GO TO 540
570     IF u=1      : GO TO 540
580     Command
590     SELECTing
600     IF sprx=2  : GO TO 580
610   END REPEAT Main
620 END DEFine
630 :
640 :
650 DEFine PROCedure INITing
660 :
670 REMark #####
680 REMark check for mode
690 REMark #####
700   IF PEEK(163892)=8 : INITtv : ELSE : INITmon
710   DIM p(16,16)
720   fa=0
730   PAPER 0,6,3
740 :
750 REMark #####

```

```

760 REMark fancy screen clear
770 REMark #####
780     FOR n= 1 TO 16
790         WINDOW n*32,n*16,256-n*16,128-n*8
800         BORDER 2,7
810         PAUSE 5
820         CLS
830     END FOR n
840 :
850 REMark #####
860 REMark Title window setup
870 REMark #####
880     BLOCK 439,20,53,6,0
890     BLOCK 442,22,47,1,2
900     BLOCK 440,20,48,2,7
910     OVER 1
920 :
930 REMark #####
940 REMark Top of the window titles
950 REMark #####
960     INK 2 : CURSOR 50,3 : CSIZE 3,1 : PRINT " SPRITE
EDITOR"
970     INK 4 : CURSOR 312,2 : CSIZE 2,0 : PRINT "Orig
by H.zENz"
980     INK 5 : CURSOR 288,11 : CSIZE 2,0 : PRINT "Mod
by L.Privett"
990     OVER 0 : INK 0
1000 :
1010 REMark #####
1020 REMark mimic window setup
1030 REMark #####
1040 REMark     BLOCK 94,9,16,26,0
1050     BLOCK 24,20,4,14,0
1060     BLOCK 20,18,6,15,7

```

```

1070  CURSOR 10, 222 : PAPER 7 : PRINT "Colour " :
PAPER 4
1080  :
1090  REMark
#####
1100  REMark #3 = first window - drawing the sprite
character
1110  REMark
#####
1120  BLOCK 126,158,16,48,0 : OPEN
#3,scr_132x162a10x42 : PAPER #3,0
1130  INK #3,7 : CLS #3 : BORDER #3,1,7 : OVER #3,-
1 : CSIZE #3,1,0
1140  :
1150  REMark
#####
1160  REMark #4 = middle window - the sprite character
set
1170  REMark
#####
1180  BLOCK 154,200,174,34,0 : OPEN
#4,scr_160x202a168x30 : PAPER #4,0
1190  CLS #4 : BORDER #4,1,7 : CSIZE #4,2,0 : PAPER
#4,2 : INK #4,7
1200  :
1210  REMark
#####
1220  REMark #5 = far right window - four original
commands
1230  REMark
#####
1240  BLOCK 134,190,358,44,0 : OPEN
#5,scr_136x190a354x42 : PAPER #5,4
1250  BORDER #5,1,7 : CLS #5 : CSIZE #5,2,0 : INK

```

```

#5,0
1260 PAPER #5,7 : PRINT #5," COMMANDS " : PAPER
#5,4 : PRINT #5
1270 PRINT #5," LOAD" : PRINT #5 : PRINT #5,"
SAVE" : PRINT #5
1280 PRINT #5," FILL" : PRINT #5 : PRINT #5,"
STORE"
1290 :
1300 REMark
#####
1310 REMark #6 = bottom window - status line
1320 REMark
#####
1330 OPEN #6,con_508x10a2x244 : PAPER #6,2 : INK
#6,4 : CLS #6
1340 CSIZE #6,2,0
1350 :
1360 REMark *****
1370 REMark draw all sprites
1380 REMark *****
1390 :
1400 FOR n=10 TO 19
1410 AT #4,(n-10)*2,7 : PRINT #4,"-";n;"-" :
SPRITE,n,38,(n-10)*5+8
1420 END FOR n
1430 AT #4,0,0
1440 FOR n=0 TO 9
1450 PRINT #4,"-0";n;"-" : PRINT #4 :
SPRITE,n,28,n*5+8
1460 END FOR n
1470 OVER #4,-1 : sprn=0 : BLOCK #4,48,10,0,0,4
1480 Arrows
1490 END DEFine
1500 :

```



```

1510 REMark #####
1520 REMark This section moves a cursor around
1530 REMark window 1 (#3) and depending on
1540 REMark the key pressed will determine
1550 REMark action. Keyboard options are displayed
1560 REMark at the bottom of the screen.
1570 REMark #####
1580 :
1590 DEFine PROCedure EDITing
1600     Message
1610     Update_Matrix
1620     pen=0 : Pend
1630     PRINT #6,"EDIT"; : Arrows
1640     BLOCK 14,10,94,222,fa
1650     x=0 : y=0 : ti=50 : Ta=0
1660     AT #3,y,x : PRINT #3,"*";
1670     Ta=KEYROW(5)
1680     t=KEYROW(1)
1690     AT #3,y,x : PRINT #3,"*";
1700 :
1710     REMark *****
1720     REMark * CTRL key pressed scrolling *
1730     REMark * through the colour selected *
1740     REMark *****
1750     IF KEYROW(7)=2
1760         fa=fa+st
1770         IF fa>7 : fa=0
1780         BLOCK 14,10,94,222,fa
1790 :
1800     REMark *****
1810     REMark ***** CTRL key not pressed *****
1820     REMark *****
1830         IF KEYROW(7)>0 : GO TO 1830

```

```

1840 END IF
1850 :
1860 REMark *****
1870 REMark The F Key Pressed invoking the
1880 REMark FILL routine
1890 REMark *****
1900 IF KEYROW (3) = 16
1910     xx=x : yy=y
1920     ToFill fa
1930     Arrows
1940     t=0
1950     x=xx : y=yy
1960 END IF
1970 :
1980 REMark *****
1990 REMark The B key pressed invoking the
2000 REMark routine for drawing a box
2010 REMark *****
2020 IF KEYROW(2)=16
2030     xx=x : yy=y
2040     ToBox fa
2050     Arrows
2060     t=0
2070     x=xx : y=yy
2080 END IF
2090 AT 0,0 : PRINT " ";
2100 AT 0,0 : PRINT t;
2110 :

```

The last part of the listing will be in the next issue
And available on the Forum website and Dilwyn's Website also.

COPY DATE for FEB/MAR 2015
ISSUE is: 5th FEB 2015

Please ensure that your copy reaches the editor on or before that date. Copy on Paper, Magnetic Media or by Email. Text & listings are preferred in ASCII format as text files, Microsoft™ Word Documents or in QL Quill formatted files. All graphic submissions in as high a resolution as possible, preferably in PNG format, other forms of submission can also be accepted, please contact the Editor for details.

All Contributions are Welcome
Copy by Email to - editor@quanta.org.uk

This magazine and all the articles etc. within it are copyright © QUANTA 2015 unless otherwise stated. Reproduction and redistribution in any form is not allowed without the express written permission of the publisher. All copyrights and trademarks are hereby acknowledged.

